DARWIN 270138

Dextrous Assembler Robot
Working with Embodied Intelligence

SEVENTH FRAMEWORK PROGRAMME
ICT Priority

Deliverable D9.2: “First year demonstrators and evaluation report”

Due date of deliverable: 31.01.2012
Actual submission date: 06.03.2012

Start date of project: 1st February 2011
Duration: 48 months

Coordinator:
Dr. Christian Eitzinger
Profactor GmbH

Lead Beneficiary: FORTH
Contributions by: IIT, CVUT

Project co-funded by the European Commission within the 7th Framework Programme

<table>
<thead>
<tr>
<th>Dissemination Level</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>PU</td>
<td>Public</td>
</tr>
<tr>
<td>PP</td>
<td>Restricted to other programme participants (including the Commission Services)</td>
</tr>
<tr>
<td>RE</td>
<td>Restricted to a group specified by the consortium (including the Commission Services)</td>
</tr>
<tr>
<td>CO</td>
<td>Confidential, only for members of the consortium (including the Commission Services)</td>
</tr>
</tbody>
</table>
Contents

1. Structure and Purpose of this Document ................................................................. 6
2. Background ............................................................................................................... 7
  2.1 The iCub humanoid robot .................................................................................. 7
  2.2 The test environment ......................................................................................... 8
  2.3 The demonstration scenario ............................................................................... 8
3. Description of Components ..................................................................................... 9
  3.1 Color-based object detection ............................................................................. 9
  3.2 3D pose estimation ............................................................................................ 11
  3.3 Motion planning and action generation ............................................................... 15
4. Component Evaluation .............................................................................................. 18
  4.1 Color-based object detection ............................................................................. 18
  4.2 3D pose estimation ............................................................................................ 22
  4.3 Motion planning and action generation ............................................................... 31
5. Conclusion ............................................................................................................... 42
References .................................................................................................................. 42
Table of figures

Figure 1: The iCub Humanoid Robot during bimanual object manipulation......................... 7
Figure 2: Examples of training images with labels for the color based detection. ................ 10
Figure 3: Segmentations in example test images. .................................................................... 10
Figure 4: Pose estimation principle. Keypoint features are detected in the image acquired online by the iCub (right) and matched with those detected off-line for the object (left). The established correspondences are used to estimate the object’s pose (R, t)................................. 11
Figure 5: Example of 3D pose estimation. The stored structure of the object (blue points) has been transformed with its estimated pose and then reprojected on the image for verification.... 14
Figure 6: Role of PMP in the territory of action “with” and ”without” movement................. 16
Figure 7: Segmentations in example images from image set 1................................................. 19
Figure 8: Examples of training images from image set 2......................................................... 20
Figure 9: Examples of successful detections for image set 2.................................................. 20
Figure 10: Examples of unsuccessful detections for image set 2............................................ 21
Figure 11. Top two rows: Sample images of a green object used to reconstruct a corresponding point cloud. The fold-up type city map on the surface supporting the object was included to increase texture that would benefit the estimation of the camera’s relative motion. Bottom row: Reconstruction result. 3D points are shown with white dots and camera viewpoints are represented with red pyramids. ........................................................................... 22
Figure 12: Side and top views of the 3D point cloud reconstructed for the green object. Notice the errors in the reconstruction showing up as points “floating” around the object......... 23
Figure 13: Sample images from iCub’s right camera that are used to evaluate pose estimation for the imaged objects at various configurations. .................................................. 24
Figure 14. Top: Presumed success per object, per feature. Column groups (1,2,3) from left to right correspond to the blue/orange/green object, respectively. Individual columns within each group from left to right correspond to ASIFT/SIFT. Bottom: Actual success, per object per feature collected from ground truth annotation of data. Graph interpretation remains the same. ................................................................. 25
Figure 15: Pose estimation results for the blue object. Blue lines correspond to results obtained with ASIFT, black to those obtained with SIFT. Points marked with red circles correspond to frames for which the corresponding pose estimation failed to provide an estimate. Top: Number of matches and outliers. Solid lines represent the matches and dashed the outliers. Bottom: First quartile of reprojection error. ........................................................................... 26
Figure 16: Pose estimation results for the orange object. Top: Number of matches and outliers. Bottom: First quartile of reprojection error. Graph interpretation is as for the blue object above .................................................................................................................. 27
Figure 17: Pose estimation results for the green object. Top: Number of matches and outliers. Bottom: First quartile of reprojection error. Graph interpretation is as for the blue and orange objects above ........................................................................................................ 28
Figure 18: Sample images and setup employed to quantitatively measure the accuracy of pose estimation. The blue object was placed in all 24 positions marked by the white squares and then photographed..................................................................................................................................... 28
Figure 19: Quantitative evaluation of 3D pose estimation. The blue line correspond to results obtained with ASIFT, black to those obtained with SIFT. Note that SIFT failed to produce a result for frame 13, hence the discontinuity in the corresponding graph. ........................................... 29
Figure 20: Example of pose estimation failure. The estimated pose for the green (bottom) object is flipped due to the fact that all matches detected for it originate from a single face. .... 30
Figure 21: Simulation of a bimanual reaching task on a 3 DoF planar arm. The figure shows the temporal evolution of both the end effector and the internal degrees of freedom (joints) from their initial configuration to their final equilibrium configuration as a result of the PMP relaxation.32
Figure 22: Bimanual coordination task of reaching and picking up two objects at the same time. Panel A: PMP network for the upper body with two target goals and a single time base generator. The network includes three modules: 1) Right arm, 2) Left arm, 3) Waist. The
a pair of nodes (‘assignment’ and ‘sum’) that allow the spread of the goal-related activation patterns. Panels B & C show the initial and the final posture of the robot and the two target objects. Panels D & E show the trajectories of the two end-effectors and the corresponding speed profiles (together with the output $\Gamma(t)$ of the time base generator). Note the end effector trajectories are straight with bell shaped velocity profiles as is well known in human reaching movements. Panel F clarifies the intrinsic degrees of freedom in the right arm-torso chain. Panel G shows the time course of the right-arm joint rotation patterns: $J_0$-$J_2$: joint angles of the Waist (yaw, roll, pitch); $J_3$-$J_9$: joint angles of the Right Arm (shoulder pitch/yaw/roll; elbow flexion/extension; wrist pronosupination pitch/yaw).  

**Figure 23:** Top panel shows the precision of reaching for 200 targets in the workspace of the robot. Pink squares: Goals, Green circles: The final position reached by the arm. 100 targets on the left half of the work space (with x-coordinate positive) were reached by the left hand and 100 targets on the right half of the work space was reached by the right arm. Bottom panel shows iCub approaching the top most object visible in the scene (i.e. user story 1).  

**Figure 24:** Several cases where synergistic coordination of the waist along with the two arms can allow reaching and grasping of objects that cannot be reached directly using the two arms. Top left panel shows two solutions to a bimanual reaching task one in which only the two arms contribute to the solution (i.e. stiff waist) and other in which the degrees of freedom of the waist is also contributes to the final solution (green). Gradual recruitment of additional degrees of freedom (waist, lower limbs, and tools) at the limit of reachability is a natural property of PMP like systems. Top left panel shows the temporal evolution of the end effector trajectories of the left and the right arm while reaching the goal. Bottom panel: A-C show an asymmetric bimanual coordination task where the goal is to grasp the cylinder placed asymmetrically with respect to the body, hence almost unreachable to the left arm. Panel A shows the final solution when only the tow arms contribute to the final solution. Panel E-F show the solution obtained by the PMP relaxation applied to the upper body in order to achieve the goal. We can observe the contribution of all three DoFs of the torso (coupled with appropriate adjustments in the right arm chain) in order to enable the left arm to cover the additional distance necessary to reach the target (along with the right arm). Panel D shows another solution where coordination of the whole upper body is essential to realize the goal. Panel E shows the final solution achieved when the goal (red cylinder) is unreachable. Note that still the PMP mechanism arrives at the best possible solution with waist movement to the allowed limits, both arms fully stretched and pointing to the direction of the goal. Such gradual degradation of performance is also a property of the PMP action system.  

**Figure 25:** A task in which only the left arm is given a target to be reached starting for an initial equilibrium configuration of the two arms. We observe that, although the goal for the right arm did not change, at the end of the task there is a slight readjustment in the posture of the right arm also. This is a kind of interference between the two PMP networks: the goal oriented force field applied to the left arm propagates to the right arm and slightly displaces/disturbs it from its target goal. Since the right arm was not supposed to move from the previously reached goal, now there is a small force field (that was earlier zero) generated autonomously in the kinematic chain of the right arm in response to this disturbance. This new force field also circulates all through the chain and the whole body reconfigures to a new posture. In other words, the external motion induced in the left arm by the force field breaks the equilibrium in the right arm and the complete system relaxes to a new solution that satisfies both goals simultaneously.  

**Figure 26:** There can also be cases where either arm can be an obstacle to the other, for example in a task where both arms have to reach targets on the other half of their workspace. Though in principle the targets can be reached, under such cases the PMP network must be augmented with a reactive mechanism that avoids self collision between the arms (one arm is an obstacle to the other).  

**Figure 27:** Goal 1 is to reach the stick placed on a table with two different wrist orientations starting from the initial condition. Goal 2 is a target that is reachable under normal conditions (without additional constraint of wrist orientation). But if there is an additional constraint of also positioning the wrist horizontally while reaching the target, based on the geometry of the arm (link
lengths) there is no solution. However as seen the PMP relaxation achieves the best possible configuration given the all the constraints (geometry of the body, reaching the goal, achieving a particular hand pose). Note that the wrist orientation is very close to zero, with the first two links fully stretched to a position that will allow the wrist to approach the goal horizontally (as much as possible). Goal 3 is an unreachable target. Also here, note that the system tries to do its best with the arm reaching as close as possible and pointing in the direction of the target. Figure 24E shows a similar task of bimanually reaching an unreachable target conducted on iCub. In this sense, gradual degradation of performance is a natural property of the attractor dynamics of the PMP mechanism.

Figure 28: iCub performing a range of manipulation tasks (reaching grasping, pushing) coordinating the upper body taking into account multiple task specific constraints (joint limits, hand pose). Note that in all these cases, reaching the goal object with specified hand pose is obligatory for successful realization of the goal.
1. Structure and Purpose of this Document

The purpose of this document is twofold. On the one hand, it will describe the basic (TE1) version of the humanoid platform which has resulted from the integration of the software components developed during DARWIN’s first year. On the other hand, it will report on the experimental evaluation of the humanoid platform using appropriate test experiments and application scenarios. The deliverable is structured around three main sections. The first of these sections introduces the humanoid robot, the environment in which it operates and the scenario chosen as the basis for the demonstration and evaluation tests. The second section describes the components that have been developed during the reported period with the aim to extend the humanoid platform’s capabilities. The third section consists of the evaluation of the components performance.
2. Background

This section describes the iCub humanoid robot and the test environment chosen for evaluation.

2.1 The iCub humanoid robot

The humanoid platform to be used for testing and evaluation is the iCub. It is the result of the project RobotCub, FP6-2004-004370. The latter is an open-source robotic platform shaped as a child humanoid robot (roughly 1m tall), with 53 degrees of freedom distributed on the head, arms, hands and legs. The iCub has been specifically designed to study manipulation, for this reason the number of degrees of freedom of the hands has been maximized with respect to the constraint of the small size. The platform is available in three copies at the IIT.

All the low-level code and documentation is provided open source by the RobotCub Consortium (http://www.robotcub.org), together with the hardware documentation and CAD drawings. The robot hardware is based on high-performance electric motors controlled by a DSP-based custom electronics. From the sensory point of view the robot is equipped with cameras, microphones, gyroscopes, position sensors in all joints, force/torque sensors in each limb. A distributed sensorized skin is also available. The hands of the iCub have five fingers and 22 joints. For practical reasons the fingers are underactuated and only 9 motors are connected to the joints. The thumb, index and middle fingers are independently controlled; this allows performing various types of grasp (full palm grasp as well as precision grip). The robot hardware is interfaced to a Pentium based machine located in the robot head. This computer is connected through Gbit Ethernet to a cluster of PCs situated nearby the robot. Cluster communication with the robot takes place through an Ethernet network. The cluster of PCs receives all sensory information from the robot (e.g. images from the cameras, and joint encoders) and can send commands to control the motor of each joint. The robot is autonomous in the range allowed by the power and Ethernet cables. Support software, as a set of libraries, is provided with the robot also according to an open-source license. These libraries include functionality for hardware control, and additional “behavioural” modules developed during RobotCub, ranging from image and sound processing to gaze and arm/hand control.

An image of the iCub robot is shown in Figure 1 below:

![Figure 1: The iCub Humanoid Robot during bimanual object manipulation.](image)
2.2 The test environment

Demonstration and testing at this phase will be done as part of the basic (TE1) environment: Testing includes simple block 3D objects, positioned in an otherwise unmodified workspace (i.e., table top). This environment primarily serves the purpose of kick-starting the development and testing of the various features for a limited set of objects. More specifically, this environment supports the development of the initial perception and action systems and facilitates testing the closed loop integration between perception and action. From the point of view of perception, the focus was to develop the first prototype object recognition and stereo reconstruction systems that feed crucial information related to object identity and spatial location (what and where) to the action system. From the action side, the goal was to set up the first prototype motion planning and action generation system that coordinates the whole upper body of iCub for a range of unimanual and bimanual actions taking into account multiple task-specific constraints. In sum, the central focus was on the creation of the first prototype perception and action systems and the achievement of robust closed loop integration between perception and action. A central feature of the developed action system is that it serves as an integrated forward/inverse model of the body and can be used both shape motor output during action execution and perform internal simulation of action to reason about feasibility and consequence of potential actions. A general computational framework for skill learning that integrates multiple streams of learning (imitation, interaction and past experience) has been developed and preliminary experiments on teaching the service robot to use common tools (and internalize the consequences of such actions) are underway. With regards to causal relations, experiments are ongoing to teach the robot the causality of ‘pushing’ various objects and learn associated forward/inverse models to support reasoning about such action in the context of a goal. These experiments are inspired by well known studies related to physical causality conducted on animals and infants (trap tube paradigm etc).

2.3 The demonstration scenario

The robot is asked to remove the top object from a stack of objects without collapsing the stack or losing the object. The single objects are colored and textured to fit the needs of the vision system. The size of the objects is such that they can be grabbed by the iCub without mechanical modifications of the robot. The vision system is allowed to gather knowledge of the employed objects in the form of prior models. The stack shall have some instability so that building the task is challenging in terms of precision of the grabbing action and the vision system. The employed objects are wooden cuboids and cylinders that measure around 5-6cm along their largest dimension and are painted with different colors and textures. Certain knowledge of their geometry has been extracted beforehand (cf. section 3.2). Additional objects creating clutter might also be present in the workspace. The stack is initially built by a human and the iCub is started at a known body posture.
3. Description of Components

The following subsections provide high level descriptions of the perceptual and action generation components that were developed during the first year of DARWIN.

3.1 Color-based object detection

In Deliverable 3.1, we discussed possible approaches to visual object detection in DARWIN. One of these approaches, developed in the first project year, was detection and recognition based on object color. Since Deliverable 3.1, this method has been improved to become a reliable and flexible component of the integrated system. Of course, detection based purely on color is not meant to be the main recognition engine of the integrated system. The typical place of this component in the system is to suggest regions of attention for more discriminative approaches, such as geometry-based detection and recognition.

Next in this section we describe the current state of the color-based detection algorithm. The task is to detect color objects in the scene. More precisely, this detection is done by segmenting the image into disjoint contiguous regions. Segmentation is done based on appearances (colors) of individual pixels and continuity prior. It is assumed that object colors are sufficiently uniform within each class and sufficiently different among different classes. Though the ideal situation is that the objects are completely texture-free, the method can deal also with textured objects, provided that the sets of colors forming individual classes are sufficiently separable.

Training. The classes are learned from examples. The input of the training phase is a set of (fully or partially) labelled images. The training is done by SVM. We use the algorithm [Franc-OCAS-2008], which is a genuinely multiclass SVM optimizer, very fast and able to cope with large numbers of samples. As the optimizer is primal, the kernelization is not possible and we construct several features from the R, G, B color channels: the feature vector is \( y = [R, G, B, RG, RB, GB, \max(R, G, B) - \min(R, G, B)] \). In this feature space, classes can be better linearly separated than just in the RGB space. The output of the learning phase is SVM weight vectors \( w \), one for each class including the background. Figure 1 shows three examples of training images with labels.
Figure 2: Examples of training images with labels for the color based detection.

**Segmentation.** After training, an unknown image is segmented using maximum-posteriori inference in a Markov random field, constructed from the image (more details can be found in Deliverable 3.1). The unary potentials in the MRF are simply scalar products $w^ty$, where $w$ is the SVM weight vector for the class and $y$ is the feature vector above. The binary potentials form the Potts model. An (approximate) MAP estimate of the labels is computed by the TRW-S algorithm [Kolmorogov-2006]. After the MRF inference, pixels from individual classes are joined into connected components, by a multi-class modification of the well-known connected component algorithm. Components smaller than a threshold are discarded via size filtering.

Figure 3 shows segmentation on example test images. In the third image, there is a slight confusion between yellow and wood because these two colors are not well separable.

Figure 3: Segmentations in example test images.
3.2 3D pose estimation

A practical problem encountered when testing the 3D reconstruction algorithms on the iCub, related to the low resolution of its cameras. More specifically, iCub’s cameras are of VGA resolution, i.e. 640X480 pixels, a fact which combined with the wide field of view of the employed lenses (about 70° horizontally and 57° vertically), results in the objects of interest being imaged with a small apparent size in iCub’s images. This, in turn, renders those images inadequate for dense stereo matching and 3D reconstruction. Due to various reasons, the replacement of the cameras with better suited ones was not possible, so the consortium has agreed to adopt an alternative approach to cope with this situation. This approach consists of off-line construction of object models based on multiple views of the employed objects, combined with on-line model matching and pose estimation. The approach is illustrated schematically in the Figure 4 and explained in more detail in subsequent paragraphs.

![Figure 4: Pose estimation principle. Keypoint features are detected in the image acquired online by the iCub (right) and matched with those detected off-line for the object (left). The established correspondences are used to estimate the object’s pose (R, t).](image)

**Offline model**

**iCub image**

**Model building.** It is assumed that an object of interest is imaged from several viewpoints and in sufficient detail. In practice, these images originate from consumer cameras with XGA resolution (i.e., 1024x768) or higher and suitable field of views. The images acquired are employed off-line to build a prior 3D model of the imaged object. For building this model, various approaches have been evaluated that consisted of using the ARC3D reconstruction web service (http://www.arc3d.be), using the Bundler structure from motion toolkit (http://phototour.cs.washington.edu/bundler/) and employing the Kinect sensor (http://www.xbox.com/en-US/kinect). ARC3D performs dense stereo matching on the images it is supplied with, producing a mesh stored in PLY

---

1 Installation of higher resolution cameras would involve undesirable hardware modifications to the iCub whereas the replacement of its lenses with longer ones would cause annoyance to other users sharing the robot.
format. Bundler relies on salient image points to compute a sparse point cloud that is saved in a text file with lots of related information. Kinect features a depth sensor which consists of an infrared laser projector combined with a monochrome CMOS camera and facilitates the capture of video data in 3D. For both ARC3D and Bundler, custom software had to be developed for interpreting their outputs and transforming them as will soon be described. ARC3D was initially chosen with the intention of permitting the construction of suitable 3D models even by users who are no vision experts. However, due to the practical problems encountered during its use (very long computing times, need for very careful image acquisition, insufficient robustness, no room for human intervention in case of failures, etc), it was decided to switch to Bundler, which has the additional advantage of not being dependent on a remote, online service. In both cases, the resulting 3D model has to be interactively cropped with a system like MeshLab (http://meshlab.sourceforge.net) in order to isolate it from the background. Kinect was eliminated from further consideration due to its unsuitable operating characteristics. More precisely, its minimum allowed distance to objects is around 60cm or more, meaning that the rather small objects of interest were not being imaged in sufficient detail. Irrespective of the tools employed and the format of their outputs, for the purposes of the remainder of this description it is enough to assume that some sort of a 3D model corresponding to the object of interest has been recovered.

Given the 3D model of an object, the next step is to identify in the images used to recover it distinct features from its surface. In the current context, the terms key and interest points are used interchangeably to refer to point-like features in an image, which have a rich local structure and are stable under local and global perturbations in the image domain (e.g. affine transformations such as stretch, squash, and skew, scale changes, rotations, zooms, and/or translations) as well as illumination and/or brightness variations. Assuming such properties, the pertinent features can be reliably computed with a high degree of reproducibility and can be matched with high accuracy. The set of interest point detectors that have been proposed in the vision literature is quite large. As the systematic comparison of [Mikolajczyk-2005] made clear, no detector outperforms all others for all types of scenes and transformations. Once interest points have been detected, a local image patch around each of them can be defined and a corresponding representation can be extracted from it. The outcome of this procedure is known as a feature descriptor or feature vector and serves as a compact representation of local image content.

Owing to its robustness to arrange of viewing transformations as well as lighting changes, our interest point detector/descriptor of choice in this work is a variant of the popular Scale Invariant Features Transform (SIFT) [Lowe-2004], called affine SIFT (ASIFT) [MorelYu-2009]. SIFT responds on textured areas and locates interest points at scale-space maxima of the Difference of Gaussian (DoG) function. ASIFT extends SIFT to make it affine invariant, as the original formulation of SIFT is only invariant with respect to scale, zoom, rotation and translation. At the cost of considerably more computation time, ASIFT simulates all poses of the original image. The simulation is controlled by two parameters, namely the horizontal and vertical angles defining the camera axis orientation. Choosing a set of discrete values for the two variables, ASIFT constructs simulations of the image to cover the whole affine space. Finally, SIFT is used to extract features from the simulated images. SIFT descriptors are made up of histograms of image gradient directions at several spatial bins, quantized to a finite set of orientations. The fact that ASIFT employs the SIFT descriptor implies that keypoints produced by ASIFT and SIFT can be directly compared, a property that is exploited in the experimental evaluation to determine the relative performance of the two detectors applied to the problem at hand. In practice, the affine invariance of ASIFT implies that
more extreme viewing distortions can be tolerated compared to the original SIFT. It has also been observed experimentally that ASIFT produces far more matchable features on a certain image compared to SIFT, being thus less dependent upon the density of texture on the object surface. Taken together, all ASIFT descriptors and the corresponding 3D coordinate vectors of their preimages obtained via reconstruction, constitute our object model and are stored in a file. Keypoints detected with the aid of Maximally Stable Extremal Regions (MSERs) [Matas-2002] that correspond to blobs of high contrast with respect to their surroundings were also tested but found to produce low numbers of features on the employed images.

**Pose estimation.** During online operation, ASIFT keypoint features are detected in an image acquired by the robot and then matched against those extracted offline for inclusion in the object’s model. The robustness of ASIFT permits the reliable identification of features that have undergone large affine distortions between the robot’s image and the model. The established correspondences are used to associate the 2D image locations of feature locations with the 3D coordinates of their corresponding points on the object’s surface. Point matching proceeds in the standard fashion for matching ordinary SIFT descriptors using a distance ratio test, as follows. Matches are identified by finding the two nearest neighbors of each keypoint from the robot image among those in the model, and only accepting a match if the distance to the closest neighbor is less than a fixed threshold of that to the second closest neighbor. This threshold can be adjusted up to select more matches or down to select only the most reliable. To make the ratio test more discriminative, a bidirectional approach is employed. Given two feature sets, the best match for the features in both feature sets is computed using the simple (unidirectional) ratio test. Then, \((f_1, f_2)\) is considered to be a match if, according to unidirectional pairings, \(f_1\) matches with \(f_2\) and \(f_2\) with \(f_1\). This approach has been found to reduce the number of false matches. Distances among SIFT descriptors are computed with the Euclidean \((L_2)\) norm. Some improvements in matching were achieved by substituting \(L_2\) with alternatives such as the Chi-squared which originates from the chi-square test statistic. This is a histogram distance that takes into account the fact that in many natural histograms, the difference between large bins is less important than the difference between small bins and should therefore be reduced. Despite that other, more expensive to compute distances such as the quadratic-Chi family [PeleWerma-2010] or the Circular Earth Movers [Rabin-2008] were found to yield even better matching results, the Chi-squared distance was eventually adopted as it offers the best performance / computational cost trade-off.

Knowledge of several 2D-3D correspondences allows the pose (i.e., position and orientation) of the object to be accurately estimated in iCub’s camera reference frame, using a combination of P3P [Haralick-1994] embedded in a robust regression framework [FischlerBolles-1981, Rousseeuw-1984] and non-linear refinement of the reprojection error. The performance of the aforementioned procedure (both in terms of accuracy and running time) can be improved by applying it only to image regions that correspond to bounding boxes of color blobs that were detected by the technique described in section 3.1.

The primary advantages of the approach described above is that it is usable irrespectively of the relative pose of the object with respect to the iCub and that the latter can easily recover from pose estimation errors as it needs to maintain very little state information. Knowledge of the camera pose enables the transformation of the model’s 3D coordinates into the frame of the iCub camera and their superposition on the image, as can be seen in Figure 5. Note that since the model’s transformed coordinates are expressed in the camera’s reference frame, they can in principle be expressed in any of
the coordinate systems employed by the iCub. A quantitative evaluation of pose estimation is presented in section 4.2.

Figure 5: Example of 3D pose estimation. The stored structure of the object (blue points) has been transformed with its estimated pose and then reprojected on the image for verification.
3.3 Motion planning and action generation

The ability to perform/learn dexterous motor skills is a fundamental requirement in order to enable Darwin robots to play, interact, learn from their world (and use such experiences to successfully perform various assembly tasks). How do goals, constraints (bodily, task specific, and environmental) and choices (redundancy) meet ‘at runtime’ in a dynamical system to give rise to spatiotemporal coordination of various degrees of freedom in a complex body (like iCub)? Perhaps this question concisely describes the central focus of our efforts in the context of this deliverable.

The problem of coordination: It is well known that along with a complex body come complex coordination and control problems. The subjective ease with which we move gracefully in constraint filled uncertain environments and perform action plans to perfection often masks the enormously complex integrative apparatus needed to spell synergy among the thousands of sensors, musculo-skeletal units and neurons that contribute to any act's planning and execution. Tasks and goals are specified at a rather high, often symbolic level (“Stack 2 cylinders”, “touch the red ball” etc.) but the motor system faces the daunting and under-specified task of eventually working out the problem at a much more detailed level in order to specify the activations which lead to joint rotations, movement trajectory in space, and interaction forces. At the same time, the solution must be compatible with a multitude of constraints: internal, external, task specific and their possible combinations. To further complicate things, there is no single solution to the problem: in fact, there may be countless ways of doing the same task. Even the task of aimlessly moving the hand from one point to another in space can be executed with a number of possible hand trajectories and timing, each trajectory could in turn be achieved in infinite ways using different combinations of joint motions at the shoulder, elbow and the wrist, and finally each joint motion can in turn be generated in infinite ways using different combinations of muscles to generate the net force at any joint. Hence arises the problem of dealing with motor redundancy and if possible, exploiting it effectively. How do humans decide what to do with their extra joints, and how should humanoid robots control all their joints in order to generate coordinated and ‘goal directed’ movement patterns?

Coordination of covert and overt actions: Mounting evidence accumulated from different directions such as brain imaging studies (Grafton 2009,), mirror neuron systems (Rizzolatti et al, 2001) and embodied cognition (Gallese et al, 2011,) generally support the idea that action ‘generation, observation, imagination and understanding’ share underlying functional networks in the brain. In general, there is growing evidence for the fact that neural circuits in the predominantly motor areas are also activated in other contexts related to ‘action’ that do not cause any overt movement. These emerging results assume significance in a project related to embodied intelligence like DARWIN where the role of coordination is not just restricted to shaping motor output during action execution but also to provide the self with information on the feasibility, consequence, understanding and meaning of ‘potential actions’. In this sense, we believe that coordination of overt action is just the tip of an iceberg: under the surface it is hidden a vast territory of actions without movements (covert actions) which are the essence of motor cognition. Hence arises the necessity to develop a sophisticated computational machinery for “Action” that is not just responsible for generation of overt movement (that iCub executes) but also acts as a core building block supporting other behaviours like
motor skill learning, imitation and goal directed reasoning which are expected to assume greater significance during the progression of DARWIN.

**Method Summary: The Passive motion paradigm (PMP):** Considering the central requirements related to ‘Action’ summarized above, a significant effort was invested in year 1 to get the first prototype of such an ‘action’ system functional in the service robot (iCub). While a detailed description of the rationale behind the approach, theory and experimental results can be both found in a recent review article (Mohan and Morasso, 2011) and D4.1 of DARWIN, here we briefly outline the general idea in order to provide context to the results presented in the component evaluation section.

![Figure 6: Role of PMP in the territory of action “with” and “without” movement.](image)

The Passive Motion Paradigm draws inspiration from the Equilibrium Point Hypothesis (EPH: Asatryan & Feldman 1965;) and is based on the theory on impedance control (Hogan, 1985). The basic idea is that actions (overt as well as covert) are the consequences of an internal simulation process that ‘animates’ the body schema with the attractor dynamics of force fields induced by the goal and task specific constraints. The internal simulation is analogous to the coordination of the movement of a marionette by means of attached strings: as the puppeteer pulls the task relevant tip to a target, the rest of its body elastically reconfigures so as to allow the tip to reach the target. The synthesis of action occurs simultaneously at multiple motor spaces (end effector, joints, muscles, tools etc), the virtual mechanical work being the structural invariant in all this. Considering that real and imagined actions turn out to be similar indeed (as evident from the emerging results from neuroscience), the proposition that even overt actions are a
product of an ‘internal simulation’ is a defining feature of PMP architecture. This is where PMP diverges from the EPH. Importantly, the internal simulation process posited by PMP offers a way to dynamically link motor redundancy with task oriented constraints ‘at runtime’, hence solving the ‘degrees of freedom problem’ without explicit kinematic inversion and cost function computation (unlike the well known optimal control framework). Operating through ‘well posed’ computations (and avoiding kinematic inversion of a redundant system) has two advantages: 1) the mechanism is not affected by the curse of dimensionality and can be easily scaled to any number of degrees of freedom; and 2) ensures that the underlying control is computationally inexpensive. Both these aspects are relevant for humans and humanoids considering the complexity of their bodies.

The mechanism is labeled ‘passive’ in line with the EPH because the equilibrium point is not explicitly specified by the brain. Instead, it just contributes to the activation of ‘task related’ force fields. The elastic reconfiguration of the body schema is a consequence of the dissemination of the goal induced force fields across the task relevant kinematic graph which characterizes the articulated structure of the human or humanoid robot. When motor commands obtained by this process of internal simulation are actively transmitted to the actuators, the robot will reproduce the same motion. Another interesting feature is that action generation networks created using PMP naturally form forward/inverse models. Hence while a trajectory of motor commands are synthesized at the intrinsic space, the forward model at the same time predicts the consequence i.e. the evolving trajectory at the end effector, a crucial piece of information for learning and improving motor performance. Simply, while the inverse model is crucial for motor control, the forward model is crucial for motor learning, their cooperation being a critical feature in the computational scheme. Hence the PMP framework both offers a unified approach to deal with both overt and covert actions, in a manner that is both computationally cheap and not affected by the complexity of the body being coordinated.
4. Evaluation

In this section, the performance of the components described in the previous section is evaluated using tests tailored to each of them. Our goal is to explore the limits of each component and gain a better understanding of the circumstances under which they fail to perform correctly.

4.1 Color-based object detection

The purpose of this section is to evaluate the color-based object detection component. We did the testing on two data sets, on the first set quantitatively, on the second set only qualitatively. For all images, the same parameters (except the SVM weights) were used, the TRW-S algorithm was run for fixed number of 5 iterations, and connected components smaller than 300 pixels were ignored.

The first data set is the one from Figure 7. It contains 45 images of size 640x480 pixels. They contain three wooden blocks, painted with different color textures. The number of color classes is 3 (blue, orange, yellow), plus background. Note that the objects have textured rather than uniform albedos (e.g., the blue object is painted with dark-blue color and then light-blue strips), but the color sets in different classes are well separable and thus “friendly” to color segmentation. The images were manually labelled, with unlabeled pixels being considered as the class “background”. Of the 45 labelled images, 8 were used for training.

All objects in all 45 images were detected, there was no false detection. Processing one image took approx 300ms on a laptop PC. The bottom rows of Figure 7 show example segmentations. The table shows the detection statistics on the whole set of 45 images. The absolute values are in pixels, the relative values (%) are with respect to the number of true positives. The last column shows the mean error of the center of gravity of the component (in pixels). The results show that despite the texture on the objects, detection works reliably. The estimates of objects’ centers of gravity are perhaps unexpectedly precise, but this is because the errors occur on the object boundary and roughly uniformly so, and hence they tend to cancel out.
The second image set consisted of 74 images, containing objects with uniform colored albedos with 4 colors (orange, red, light blue, green). Figure 8 shows 3 example training images (the dark blue box in the top-right corner was not included). Figure 9 shows examples of successful detections. Note that the holes are accurately segmented. However, Figure 10 shows that for the screw and the screwdriver, detection partially or completely failed. This is because the objects are relatively small (and hence the “shrinkage bias” of the Potts model has a relatively large effect) and, most importantly, the colors are not well separable in the feature space (orange from red, dark blue from background). These limits of the method are not unexpected.
Figure 8: Examples of training images from image set 2.

Figure 9: Examples of successful detections for image set 2.
Figure 10: Examples of unsuccessful detections for image set 2.
(left: the blue top undetected; middle: complete miss; right: the red top undetected)
4.2 3D pose estimation

The performance of the pose estimation module was assessed with the aid of several experiments. The performed experiments employed as test objects wooden blocks of different shapes painted with lines to create texture. Initially, each object was photographed individually in about 40 images as a camera circumnavigated twice around it (see Figure 11) and then the acquired images were used to estimate the interframe camera motion and recover a corresponding 3D point cloud via triangulation (see Figure 12).

Figure 11. Top two rows: Sample images of a green object used to reconstruct a corresponding point cloud. The fold-up type city map on the surface supporting the object was included to increase texture that would benefit the estimation of the camera's relative motion. 
Bottom row: Reconstruction result. 3D points are shown with white dots and camera viewpoints are represented with red pyramids.
Following the creation of suitable object models, three test objects were placed in iCub’s workspace at arbitrary configurations with varying degrees of occlusions and distances from iCub’s head. Each such configuration was captured from iCub’s right camera, yielding a test dataset consisting of 46 images. Representative images from this dataset are shown in Figure 13. The following table lists some properties of the dataset related to the number of objects appearing in the images and the presence of occlusions. Each of the 46 images was used to estimate the pose of all objects appearing in it and the results gathered are presented below.

<table>
<thead>
<tr>
<th>#Images containing two objects</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>#Images containing three objects</td>
<td>43</td>
</tr>
<tr>
<td>#Images with occlusions</td>
<td>23</td>
</tr>
</tbody>
</table>

Properties of the dataset used to evaluate pose estimation.
Aiming to demonstrate the improvements brought about by ASIFT, it was decided to include in the following results from two sets of pose estimation experiments, namely one set based on features detected with ASIFT and one with those obtained with plain SIFT. The mean reprojection errors in pixels for both feature detectors and all three different objects are summarized in the following table. According to it, SIFT seems to perform better, yielding smaller reprojection error. Note, however that such errors are only meaningful when the pose estimation algorithm has succeeded in finding a solution.

<table>
<thead>
<tr>
<th></th>
<th>ASIFT</th>
<th>SIFT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blue</td>
<td>1.021434</td>
<td>0.692509</td>
</tr>
<tr>
<td>Orange</td>
<td>2.374598</td>
<td>1.073416</td>
</tr>
<tr>
<td>Green</td>
<td>1.035596</td>
<td>0.827225</td>
</tr>
</tbody>
</table>

Reprojection errors in pixels for the three test objects.

Considering that the reprojection error alone is not always a trustworthy measure of accuracy, the following graphs in Figure 14 illustrate for each of the three objects, the success rate of its pose estimation as reported by the estimation algorithm (top) as well as the actual success rate as determined by a human through visual inspection of the results (bottom). As can easily be seen from the bottom graph, pose estimation using ASIFT performs correctly from about 80% to 90% of the time, depending on the actual object. In the case of SIFT, the corresponding rates drop from about 80% to 30%. Especially for the orange object, a difference of performance in favour of the ASIFT is evident.

Figure 13: Sample images from iCub's right camera that are used to evaluate pose estimation for the imaged objects at various configurations.
Figure 14. Top: Presumed success per object, per feature. Column groups (1,2,3) from left to right correspond to the blue/orange/green object, respectively. Individual columns within each group from left to right correspond to ASIFT/SIFT. Bottom: Actual success, per object per feature collected from ground truth annotation of data. Graph interpretation remains the same.

The next group of graphs show for each object, the number of matches determined via matching SIFT descriptors that were subsequently fed to pose estimation (solid lines) plus the number of outliers detected during this estimation (dashed lines). Outliers typically correspond to mismatched points that should be discovered and eliminated in order to prevent them from influencing the computed pose estimate. These graphs confirm our expectation that the number of matches decreases with the distance from the camera and the amount of occlusions. In addition to matches, graphs for the first quartile of the reprojection error are included. In descriptive statistics, the first quartile (designated q1) of a set of values is the point dividing these values into two groups with the first group containing a fourth of the population being sampled. In simpler words, the first quartile cuts off the lowest 25% of data, i.e. corresponds to the 25th percentile. Figure 15, Figure 16 and Figure 17 respectively illustrate the performance of the pose...
estimation for the blue, orange and green objects. As it can be observed in the graphs, peaks in the reprojection error correspond to frames where the number of outliers represents a high proportion of the total matches found. As can be seen from the plots, the reprojection error for the orange object is higher to those for the blue and green objects. As can be seen from Figure 16 top, this can be attributed to the higher fractions of outliers in the sets of matches obtained for it.

Figure 15: Pose estimation results for the blue object. Blue lines correspond to results obtained with ASIFT, black to those obtained with SIFT. Points marked with red circles correspond to frames for which the corresponding pose estimation failed to provide an estimate. Top: Number of matches and outliers. Solid lines represent the matches and dashed the outliers. Bottom: First quartile of reprojection error.
Figure 16: Pose estimation results for the orange object. Top: Number of matches and outliers. Bottom: First quartile of reprojection error. Graph interpretation is as for the blue object above.
Figure 17: Pose estimation results for the green object. Top: Number of matches and outliers. Bottom: First quartile of reprojection error. Graph interpretation is as for the blue and orange objects above.

The average running time of the ASIFT approach was 11.9 seconds on a desktop Windows 7 machine. Running times are of course dependent on the size of the employed bounding boxes and the density of the object’s texture, which is reflected in a high standard deviation that was measured to be 5.6 seconds. The corresponding figures for SIFT were 1 second for the average and 0.35 for the standard deviation.

Figure 18: Sample images and setup employed to quantitatively measure the accuracy of pose estimation. The blue object was placed in all 24 positions marked by the white squares and then photographed.

Aiming to quantitatively assess the accuracy of pose estimation, an experiment with known ground truth for the object pose was performed next. A sheet of paper with known marked positions was printed and put on the working space in front of the iCub, as illustrated in Figure 18. Then, the blue object was photographed as it was placed to all marked positions in turn and the resulting images were used to estimate the pose of the object. Using the known dimensions of the marked rectangles, the pose of the black sheet of paper was estimated as described in [Zhang-2000] and the known size of the blue object was employed to predict the coordinates of its center of gravity, which played the role of ground truth for each image. Strictly speaking, the predicted centers of gravity do not correspond to the ground truth as they are produced by a process susceptible to measurement error. Nevertheless, they do provide an approximation of the true coordinates of the object’s center of gravity and, considering the practical difficulty of collecting ground truth data with independent means, suffice for basing a comparison.
Using the estimated camera poses, the coordinates of the object's center of gravity in each of the 24 images were then inferred. This is achieved simply by transforming the center of gravity stored with the reconstructed model using the estimated 3D pose. Finally, the distance of the inferred centers of gravity from the ground truth was computed. The results of this comparison are shown in Figure 19, from which can be seen that this error is less than 2cm in most cases.

![Figure 19: Quantitative evaluation of 3D pose estimation.](image)

The experimental evaluation has confirmed that the most common failure mode for the pose estimation module relates to the lack of sufficiently many matched features between the robot's image and the stored 3D model. The typical reasons for obtaining too few image matches are the large distance of the object from the camera and poor scene illumination. The limited numbers of detected points is also related to the moderate resolution and wide field of view of the employed cameras. Also, the pose of objects that are too close and below the camera is difficult to estimate as their faces undergo significant perspective distortions which complicates matching. A somewhat less frequent failure mode occurs when all available matches originate from the same planar face of the object. In this case, the available constraints are insufficient for uniquely determining pose and the estimated pose might be “flipped”, as shown in Figure 20. The texture of the employed objects also has a certain impact on the robustness of its pose estimation. Clearly, the denser the texture of the object, the more keypoints can be recognised in its images, imposing more constraints on pose estimation. Higher spatial frequency textures are also more favourable for the pose estimation as they give rise to more distinctive keypoints.
Figure 20: Example of pose estimation failure. The estimated pose for the green (bottom) object is flipped due to the fact that all matches detected for it originate from a single face.


### 4.3 Motion planning and action generation

In the context of this deliverable, the evaluation of the PMP architecture is focussed on the subtask of coordinating the upper body of iCub (30 degrees of freedom) to perform various unimanual and bimanual actions simultaneously taking into account a range of bodily and task related constraints depending on the objects being manipulated. In the subsections that follow, analysis of the performance of the system is presented in tasks of gradually increasing complexity:

- **Analysis of a bimanual reaching task** (i.e. reach two different objects at the same time with the right and the left hand respectively) that focuses on the end effector trajectories, trajectories in the intrinsic space, analysis of the timing and velocity profiles.
- **Evaluation of the precision of reaching** in the workspace relevant for the task environment of un-stacking objects.
- Performance of the system while reaching and picking up objects almost at the *limit of reachability*, asymmetric bimanual coordination tasks in which cases synergistic use of the additional degrees of freedom in the waist is necessary to realize/almost realize the goal.
- **Incorporation of multiple constraints** like joint limits, achieving a specific hand pose while picking up objects (MECCANO screw or stick etc).

### Analysis of a bimanual reaching task

In this section we evaluate the end effector trajectories, velocity profiles, timing signals, temporal evolution of degrees of freedom in the intrinsic space (upper body) as the PMP network generates a goal directed action using both arms. Figure 22 a shows the PMP network for coordinating the upper body of iCub that is relevant for all tests related to this deliverable. To provide context to the graphs we briefly summarize what the nodes mean in this fully connected network and how information flows through them. The description applies to any PMP network in general (whole body with lower limbs, additional tools coupled to various end effectors etc). As seen, PMP networks operate concurrently in multiple motor spaces (tool, end effector, arm joints, waist), each motor space consisting of a generalized displacement (blue) and force node (pink). The scalar work ($dx \cdot df$) is the structural invariant across motor spaces. There are two kinds of links between the nodes. The vertical links describe the elastic causality of the coordinated system and are characterized by stiffness ($K$) and admittance matrices ($A_j$ and $A_f$). Horizontal links connecting different motor spaces represent the geometric causality and are characterized by the Jacobian matrices ($J$). In complex kinematic structures, characterized by multiple serial or parallel connections, the sum and assignment nodes are used to add or assign displacements and forces to different connecting elements of the kinematic chain (in this case the connectivity of the two arms with the waist). For the simplest case of bimanual reaching, the goal generates attractive force fields $F_R=K(x_{Goal.R}-x_{ini})$ and $F_L=K(x_{Goal.L}-x_{ini})$ applied at the respective end effectors. These force fields are mapped from extrinsic to intrinsic space by means of the mapping $T_R=J_R^T F_R$ and $T_L=J_L^T F_L$ respectively, hence yielding attractive torque fields in the joint spaces of both arms. These torque fields induce a coordinated motion of all the joints in the arm and the waist according to the admittance matrix $A$. Explicit manipulation of the admittance parameters alters the contributions of different joints in the arm and torso towards the overall reaching movement. The motion of the joints now, determines the
motion of the right and left end effectors according to the forward relationship $\dot{x} = J\dot{q}$. Ultimately, the motion of the iCub kinematic chain evoked by the application of a goal target is equivalent to integrating non-linear differential equations that, in the simplest case (i.e. no additional task specific constraints), takes the following form: $\dot{x} = \Gamma(t) J A J^T K (x_{Goal} - x)$ where, $x$ is the end-effector position, $K$ is the virtual stiffness in the end effector space, $J$ the Jacobian and $A$ the admittance in the joint and waist space. The parameter $\Gamma(t)$ is a timing signal that implements the terminal attractor dynamics (Zak, 1988) that allows reaching the goal target in a finite specified time, synchronization between multiple arms (for example consider a task of bimanually transporting an object).

In order to visualize of the incremental reconfiguration of the body from the initial configuration to the final equilibrium configuration (triggered by the pull of the goal), Figure 21 shows a simulation of bimanual reaching task on a 3 DoF planar arm. In this case, since the targets were symmetrically located the motion of the two arms from the initial configuration to the final configuration is also symmetric. The end effector trajectories are also approximately straight with bell shaped velocity profiles (Figure 22 E) as has been observed in human reaching movements (Morasso, 1981). Going beyond the initial 3DoF planar arm simulation, Figure 22 B-G presents an analysis of the same bimanual reaching task using all the 17 degrees of freedom of the iCub upper body. In general, we remark that since PMP networks always operate through ‘well posed’ computations, they do not suffer from the curse of dimensionality and can be scaled easily to coordinate bodies with arbitrary complexity and redundancy (Mohan and Morasso, 2011).

![Figure 21: Simulation of a bimanual reaching task on a 3 DoF planar arm.](image)

The figure shows the temporal evolution of both the end effector and the internal degrees of freedom (joints) from their initial configuration to their final equilibrium configuration as a result of the PMP relaxation.
Figure 22: Bimanual coordination task of reaching and picking up two objects at the same time. Panel A: PMP network for the upper body with two target goals and a single time base generator. The network includes three modules: 1) Right arm, 2) Left arm, 3) Waist. The dimensionality of $J_R$ & $J_L$ is 3×10 (this includes the 7 DoF's of the arms and the 3 DoF's of the waist). The dimensionality of $A_j$ is 7×7 and of $A_T$ is 3×3. The three sub-networks interact through
a pair of nodes (‘assignment’ and ‘sum’) that allow the spread of the goal-related activation patterns. Panels B & C show the initial and the final posture of the robot and the two target objects. Panels D & E show the trajectories of the two end-effectors and the corresponding speed profiles (together with the output $\Gamma(t)$ of the time base generator). Note that the end-effector trajectories are straight with bell shaped velocity profiles as is well known in human reaching movements. Panel F clarifies the intrinsic degrees of freedom in the right arm-torso chain. Panel G shows the time course of the right-arm joint rotation patterns: $J_0$-$J_2$: joint angles of the Waist (yaw, roll, pitch); $J_3$-$J_9$: joint angles of the Right Arm (shoulder pitch/yaw/roll; elbow flexion/extension; wrist pronation/supination pitch/yaw).

**Evaluation of the precision of reaching**

In the previous subsection we presented an analysis of the qualitative characteristics of motion generated using the PMP architecture for iCub upper body coordination. The analysis was restricted to single bimanual reaching task. In this section we evaluate the precision of reaching in general as the targets vary from one location to another in the workspace of the robot.

Figure 23 shows a set of 200 goal targets in the work space of the robot (shown in pink) and the final position of the end effector after the movement to reach the goal target (green circles). 100 targets on the left half of the work space (with x-coordinate positive) were reached by the left hand and 100 targets on the right half of the work space was reached by the right arm. In general, in the work space with volume of $44\times35\times30$ cm$^3$ (along x, y and z axes respectively) in front of the robot the mean precision of reaching for both arms is approximately in the range of 4mm with a variance of 2 mm. Considering that the dimensions of the objects manipulated in user story 1 are blocks of approximate volume $5$cm$^3$ the performance of the action system is highly efficient in terms of supporting reachability and graspability of objects on interest. Even for objects in the MECCANO 2+ kinds play tool kit (screws, screw driver, blocks etc) this level of performance is reasonable.
**Figure 23:** Top panel shows the precision of reaching for 200 targets in the workspace of the robot. Pink squares: Goals, Green circles: The final position reached by the arm. 100 targets on the left half of the work space (with x-coordinate positive) were reached by the left hand and 100 targets on the right half of the work space was reached by the right arm. Bottom panel shows iCub approaching the top most object visible in the scene (i.e. user story 1).

The smallest and most challenging object, i.e. the screw, has a diameter of 2.5cm (this is designed for 2-3 year old children and suited for small robot like iCub). Such objects also can be certainly reached, though increasing success of grasp functionally needs tighter integration with other sensory streams like touch (using touch sensors mounted in the finger tips and palm) to provide feedback for minor adjustments at the edge of physical interaction (where most often vision is partially or completely occluded). Work is ongoing towards tighter integration of touch with the PMP system to support efficient manipulation of tiny objects.

**Synergistic use of ‘Left arm-Torso-Right arm’ for Targets at the limits of reachability**

In the previous two sections we evaluated the qualitative characteristics of action generated by the PMP system and the precision of such actions in the reachable bimanual workspace of the robot. In this section we move further to tasks where objects are placed at the limit of reachability for both arms (for example very far) or for one of the two arms (for example an asymmetric bimanual reaching task where both arms have to reach the same target that is placed). In such cases it is possible to incrementally recruit and exploit the additional 3 degrees of freedom in the trunk to either realize the goal or move as close as possible. Figure 24 shows several cases where synergistic coordination of the waist with the two arms can allow reaching and grasping of objects that cannot be reached directly using the two arms.
Figure 24: Several cases where synergistic coordination of the waist along with the two arms can allow reaching and grasping of objects that cannot be reached directly using the two arms. Top left panel shows two solutions to a bimanual reaching task one in which only the two arms contribute to the solution (i.e. stiff waist) and other in which the degrees of freedom of the waist is also contributes to the final solution (green). Gradual recruitment of additional degrees of freedom (waist, lower limbs, and tools) at the limit of reachability is a natural property of PMP like systems. Top left panel shows the temporal evolution of the end effector trajectories of the left and the right arm while reaching the goal. Bottom panel: A-C show an asymmetric bimanual coordination task where the goal is to grasp the cylinder placed asymmetrically with respect to the body, hence almost unreachable to the left arm. Panel A shows the final solution when only the tow arms contribute to the final solution. Panel E-F show the solution obtained by the PMP relaxation applied to the upper body in order to achieve the goal. We can observe the contribution of all three DoFs of the torso (coupled with appropriate adjustments in the right arm chain) in order to enable the left arm to cover the additional distance necessary to reach the target (along with the right arm). Panel D shows another solution where coordination of the whole upper body is essential to realize the goal. Panel E shows the final solution achieved when the goal (red cylinder) is unreachable. Note that still the PMP mechanism arrives at the best possible solution with waist movement to the allowed limits, both arms fully stretched and pointing to the direction of the goal. Such gradual degradation of performance is also a property of the PMP action system.

It is worth noting that the motions of the two arms are not totally independent when the trunk is compliant: there is a propagation of the force field applied on one hand to the other arm and vice-versa. Such interference occurs because the waist is serially connected to both arms and responds to the pull of either arm. Figure 25 shows a result an example task in which only the left arm is given a target to be reached starting for an initial equilibrium configuration of the two arms (shown in red). We observe that, although the right target did not change, in the final solution there is a slight readjustment in the posture of the right arm also. This is a kind of interference between the two PMP sub networks of the right and the left arm though the waist: the goal
oriented force field applied to the left arm propagates to the right arm and slightly displaces/disturbs it from its target goal. Since the right arm was not supposed to move from the previously reached goal, now there is a small force field generated autonomously in the kinematic chain of the right arm in response to this disturbance. This new force field also circulates all through the chain and the whole body reconfigures to a new posture. In other words, the external motion induced in the left arm by the force field breaks the equilibrium in the right arm and the complete system relaxes to a new solution that satisfies both goals simultaneously.

Figure 25: A task in which only the left arm is given a target to be reached starting for an initial equilibrium configuration of the two arms. We observe that, although the goal for the right arm did not change, at the end of the task there is a slight readjustment in the posture of the right arm also. This is a kind of interference between the two PMP networks: the goal oriented force field applied to the left arm propagates to the right arm and slightly displaces/disturbs it from its target goal. Since the right arm was not supposed to move from the previously reached goal, now there is a small force field (that was earlier zero) generated autonomously in the kinematic chain of the right arm in response to this disturbance. This new force field also circulates all through the chain and the whole body reconfigures to a new posture. In other words, the external motion induced in the left arm by the force field breaks the equilibrium in the right arm and the complete system relaxes to a new solution that satisfies both goals simultaneously.
Figure 26: There can also be cases where either arm can be an obstacle to the other, for example in a task where both arms have to reach targets on the other half of their workspace. Though in principle the targets can be reached, under such cases the PMP network must be augmented with a reactive mechanism that avoids self collision between the arms (one arm is an obstacle to the other).

While Figure 25 showed a very subtle effect of interference that caused autonomous reconfiguration of body posture to counteract its effect, there can also be cases where either arm can be an obstacle to the other, for example in a task where both arms have to reach targets on the other half of their workspace (Figure 26). In such cases the PMP network must be augmented with a mechanism that avoids collision between the arms (one arm is an obstacle to the other). Though in principle it is possible to reach such targets using PMP as seen in the simulation, such cases are rare for the scenarios relevant so far. However, reactive mechanism to avoid self collisions will be incorporated in the near future to make the PMP system robust to generate actions under such conditions.

Incorporating multiple Constraints
Motor commands needed to perform even seemingly simple actions like lifting a cup of coffee are also dependent on several variables both internal and external to the body like the current state of the arm (joint angles) and orientation of the body, range of motion for the joints, range of torques for the actuators, geometry of the task (shape of the cup), obstacles in the environment to list a few. We effortlessly modify the orientation of our bodies / arms based on the nature of the physical objects we interact with or based on the nature of the task being performed. Proper orientation often
increases the chances of successful execution of a task. This subsection presents examples of tasks related to user story 1 and 3 that incorporate multiple task related constraints by composing attractor landscapes that combine multiple force fields in different reference systems. Typical internal constraints are related to joint limit avoidance, i.e. keeping each DoF inside a given range of motion:

\[ \{ q_{i,\text{min}} < q_i < q_{i,\text{max}}, \ i = 1, n \} \]

They can be implemented by adding an elastic force field in the joint space, with a nominal equilibrium for example, in the middle of the range of motion of each joint. Reaching the goal target with a specific hand pose is a particularly important prerequisite for successful grasping and manipulation of various objects/tools placed on a table in front of the robot. This is achieved by extending the basic PMP network with three superimposed force fields that shape the spatiotemporal behaviour of the system: 1) To the end-effector (to reach the target); 2) To the wrist (to achieve the specified hand pose); 3) Force field in joint space for joint limit avoidance. A single timing signal \( \Gamma(t) \) synchronizes all the three relaxation processes to equilibrium. The theoretical details are explained in greater detail in D.4.1 and in Mohan and Morasso (2011).

Figure 27 and Figure 28 present the analysis of the performance of the basic PMP network coordinating the iCub upper body (Figure 21) to generate actions that include combinations of other task specific constraints that have to be additionally satisfied while realizing the goal. To help visualization of the temporal evolution of the motion from the initial configuration to the goal, Figure 27 shows the simulation of the behaviour for a 3DoF arm in different cases. Goal 1 is to reach the stick placed on a table with two different wrist orientations starting from the initial condition. As shown in Figure 28 such requirements are highly relevant while manipulating objects most of the objects in the real world scenario with iCub. Goal 2 is a target that is reachable under normal conditions (without additional constraint of wrist orientation). But if there is an additional constraint of also positioning the wrist horizontally while reaching the target, based on the geometry of the arm (link lengths) there is no solution. However as seen in Figure 28 the PMP relaxation achieves the best possible configuration given the all the constraints (geometry of the body, reaching the goal, achieving a particular hand pose). Note that the wrist orientation is very close to zero, with the first two links fully stretched to a position that will allow the wrist to approach the goal horizontally (as much as possible). Goal 3 is an unreachable target. Also here note that the system tries to do its best with the arm reaching as close as possible and pointing in the direction of the target. Figure 24E shows a similar task of bimanually reaching an unreachable target conducted on iCub. In this sense, gradual degradation of performance is a natural property of the attractor dynamics of the PMP mechanism.

Figure 28 shows results of iCub performing different manipulation actions on objects satisfying additional constraints of approaching the goal with a desired hand pose to aid further manipulation and an internal constraint of keeping the motion of all intrinsic joints as close as possible to the mid range of allowed range of motion. While a quantitative analysis is dependent on number of factors like location of the target that influences the reachability and the desired hand pose (see case of goal 2), the strength of the individual force fields, other constraints also acting simultaneously, we remark that the PMP relaxation attempts to find a solution that is the best possible balance among multiple requirements (at the same time exploiting the redundancy in the body). In the reachable workspace of the robot the performance of the system is close to 100%.
Figure 27: Goal 1 is to reach the stick placed on a table with two different wrist orientations starting from the initial condition. Goal 2 is a target that is reachable under normal conditions (without additional constraint of wrist orientation). But if there is an additional constraint of also positioning the wrist horizontally while reaching the target, based on the geometry of the arm (link lengths) there is no solution. However as seen the PMP relaxation achieves the best possible configuration given the all the constraints (geometry of the body, reaching the goal, achieving a particular hand pose). Note that the wrist orientation is very close to zero, with the first two links fully stretched to a position that will allow the wrist to approach the goal horizontally (as much as possible). Goal 3 is an unreachable target. Also here, note that the system tries to do its best with the arm reaching as close as possible and pointing in the direction of the target. Figure 24E shows a similar task of bimanually reaching an unreachable target conducted on iCub. In this sense, gradual degradation of performance is a natural property of the attractor dynamics of the PMP mechanism.
Figure 28: iCub performing a range of manipulation tasks (reaching, grasping, pushing) coordinating the upper body taking into account multiple task specific constraints (joint limits, hand pose). Note that in all these cases, reaching the goal object with specified hand pose is obligatory for successful realization of the goal.
5. Conclusion

This deliverable has presented the software components that have been developed during DARWIN’s first year and has reported on their experimental evaluation on the iCub humanoid platform.

References


